

2000162
P-40
NASA Research Cooperative Agreement NCC1-122

AUTOMATIC DETERMINATION OF FAULT EFFECTS
ON AIRCRAFT FUNCTIONALITY

Semi-Annual Progress Report
February 17 - August 17, 1989

Stefan Feyock

Computer Science Department
College of William & Mary
Williamsburg, VA 23185

(NASA-CR-181352) AUTOMATIC DETERMINATION OF
FAULT EFFECTS ON AIRCRAFT FUNCTIONALITY
Semiannual Progress Report, 17 Feb. - 17
Aug. 1989 (College of William and Mary)
40 p

N91-10039

Unclass
0280462

CSCL 01C 63/05

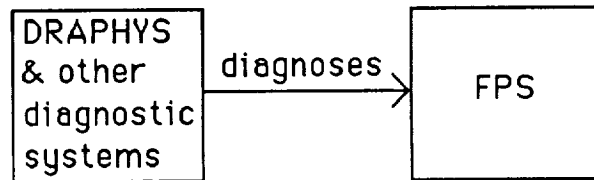
INTRODUCTION

The research described in this report was performed in conjunction with the Intelligent Cockpit Aids (ICAT) project conducted by the Vehicle Operations Branch at NASA Langley Research Center. The goal of this project is to develop artificial intelligence (AI) techniques and systems to assist flight crews in the performance of their tasks. Such assistance can become particularly crucial when malfunctions occur; a significant portion of the project is accordingly devoted to the development of tools that will help flight crews cope with system faults.

When an in-flight malfunction occurs, the first priority after stabilizing the aircraft is to determine the nature of the fault. Much of the prior work on this project accordingly focused on fault diagnosis. A number of approaches were taken, including traditional rule-based expert systems, model-based performance monitoring [Sch], and model-based diagnosis [Abb]. These systems have achieved a significant level of success, and are able to supply plausible diagnoses for a wide variety of fault symptoms.

Once the nature of the fault has been established, the flight crew must determine how to deal with it. This task involves determining the effect of the malfunction on the aircraft's performance, both momentary and in the future. The latter determination involves developing a *prognosis* of the effects of the fault, including prediction of possible propagation to other aircraft components. The research described in this report had the goal of developing AI-based techniques to assist in the development of such prognoses.

Since the above-mentioned diagnostic dealt quite successfully with the task of determining the nature of the malfunction, it was possible to assume, for the purposes of prognosis development, that the output of diagnostic systems such as DRAPHYS would form the input to our prognosis-development system. The goal of our project was thus the automatic construction of answers to the question: *what will be the effects of this fault, and how will it propagate?* The following diagram illustrates this organization; FPS denotes the Fault Propagation System.



How this question is answered depends on the amount of information available. The least amount of information that can be available about a fault is the mere fact that it exists, e.g. "fuel system malfunction". If only such minimal information is available, only qualitative answers reflecting component status can be derived. For example, given a diagnosis of *fuel system malfunction*, the system can produce the following prognosis:

propagation from fuel system to combustor

propagation from combustor to turbine_1

etc.

The Draphys system reasons on the basis of such minimal information to produce diagnoses, but can be run "forward" to produce predictions of future behavior.

If information about changes in system parameters is available, more precise qualitative answers can be produced. For example:

Diagnosis:

fuel system malfunction: decreased fuel flow

Prognosis:

propagation from fuel system to combustor

Effect: burn rate decreased

propagation from combustor to turbine_1

Effect: rotation speed decreased

etc.

This kind of reasoning requires qualitative knowledge of how parameters that represent component and process measurements affect each other. The work of Kuipers [Kui] typifies this approach: constraints among parameters are used to produce qualitative simulations of the system in question. These constraints may be

arithmetic, e.g. $A = B + C$

functional: $Y = M^+[X]$ (resp. $Y = M^-[X]$) denotes that Y is a monotonically increasing (resp. decreasing) function of X

derivative: $Y = dX/dt$

Such qualitative information is frequently available. It is worth noting, however, that in many real-life situations the relative time scale of events is crucial. It is necessary, for example, to distinguish between a tire blowout and a slow leak, since the response to the two situations differs radically; most current qualitative reasoning systems do not, however, have adequate facilities for making such distinctions.

In the ideal situation, of course, sufficient information is available to permit the derivation of quantitative answers such as

fuel system malfunction has reduced
engine performance by 70%

Impact on thrust: reduced by 80%

Impact on fuel consumption: increased by 15%

Impact on range: reduced by 50%

etc.

The research being reported has focused on reasoning based on quantitative models, in particular models of continuous systems. It goes almost without saying that such models have many advantages. After hundreds of years of development this notation is extremely concise, powerful and expressive, allowing analysis as well as numeric simulation. Furthermore, qualitative statements such as "burn rate decreased" may be minimally useful, in that it is not possible to determine the appropriate course of action on the basis of such information: divert to alternate airport? brace for impending crash? Quantitative information is needed to distinguish between such alternatives.

In defense of qualitative reasoning it must be stated that qualitative information is often valuable, especially in the context of highly complex systems where fault ramifications are not obvious. In fact, it is clear that in an emergency situation the personnel involved will lack the leisure required to study reams of qn simulation output, and that only qualitative condensations of the data will be of use. Furthermore, the generally chaotic nature of failure situations makes it unlikely that precise information will be available. The automated system must do the best it can in this context, but *no better* : producing quantitative data of spurious precision from imprecise inputs is worse than useless. It is expected that future work will integrate qualitative reasoning to a greater extent.

In the final analysis, the motivation for persevering in the use of quantitative model technology even in the absence of precise information

was the superior ontology offered by system dynamics models, particularly bond graphs. This ontology is discussed in detail below.

QUANTITATIVE PROGNOSTICATION

Traditional expert system technology is based on rules such as

if symptom a then fault b

if fault x then symptom y

together with inference processes of greater or lesser sophistication that reason on the basis of these rules. This approach works well for *anticipated* faults: those which have been experienced previously either in the field, in computer simulations, or predicted by human experts. For this reason a traditional expert system should form the first stage of the diagnostic/prognostic system in question, performing the function of dealing with familiar situations.

It is inevitable, however, that there will be a residue of occurrences with which such an expert system cannot cope, since these occurrences are either unforeseen, or correspond to novel constellations of familiar faults. Traditional expert systems are notorious for their graceless degradation in unforeseen circumstances; a different approach is required. It is clear that to deal with unforeseen circumstances we need to reason from "first principles", i.e. from models of the structure and functionality of the system. DRAPHYS deals with unanticipated faults by employing such model-based reasoning, and our prognosticator, which we have named FPS, (Functional Propagation System) follows and extends this approach.

Models and Model-based Reasoning

We have stated that it is necessary to have a model of the subject system in order to reason about unanticipated phenomena involving that system. If we accept the definition "A is a model of B for C if C can use A to obtain answers about B", the statement is true by definition. Unfortunately this definition is overly general: a traditional expert system certainly qualifies, as does flipping a coin. What is required for our purposes is a formal system whose objects and relationships map onto the real-world objects and relationships of interest. The objects of interest are the physical components of the machines and systems undergoing failure; the relevant relationships include causality, *is_a*, *part_of*, among others. Any model we employ must reflect these objects and relations.

Models of the engineered systems of interest to us, particularly airplane systems and subsystems, are widely available: the reliability requirements of aviation are such that it is not feasible to use components whose operation is poorly understood. "Well-understood" in this context means that mathematical and/or simulation models of the (sub)system exist or can be formulated. In addition, it is generally possible to predict the behavior of composite systems if their components are well understood. Unfortunately for us, however, mechanical failures produce systems undreamt of in engineers' imaginings, and it is about such pathological components that we must reason. Thus the question is:
how do we obtain a model of the faulty system?

An obvious answer is to update the model of the intact system in some fashion. This may not be possible, however, since the intact model may not reflect the causal mechanisms involved in the modeled system at all: it may be no more than a response surface. Models consisting of regression lines fitted to data derived from the physical system exemplify this situation.

For our application domain it is usual for models of dynamical physical systems to be based on differential equations, which reflect at least some aspects of the physical system. A typical example is provided by linear

harmonic oscillators such as the mass/spring/damper system:

$$m \cdot x'' + d \cdot x' + k \cdot x = F(t)$$

Dynamical systems in general can be described by *state equations*, which have the canonical form

$$x'_1 = F_1(x_1, \dots, x_n, u_1, \dots, u_r)$$

...

$$x'_n = F_n(x_1, \dots, x_n, u_1, \dots, u_r)$$

The linear harmonic oscillator, for example, can be put in this form by introducing a new variable $u = x'$; the equation then becomes

$$u' = (F(t) - d \cdot u - k \cdot x) / m$$

$$x' = u$$

Later we will discuss methods for systematically deriving canonical-form state equation sets whose variables have a closer correspondence to the underlying physical quantities.

Models built from differential equations are based on what may be termed an *adjective-oriented ontology*. Ontology, a philosophical term denoting the nature of being, refers in this context to the basic building blocks available to the modeler for model construction. It is evident that components and processes are not explicitly represented in the ontology underlying state equations. Instead, the equations express constraints among *attributes* of components: the *massiveness* of an object, the *resistivity* of a damper, the *stiffness* of a spring. The components to which these attributes belong have been abstracted away; the numbers, types, and interconnections of these components can not, in general, be recovered from the state equations.

The fact that components are not explicitly represented in an adjective-oriented ontology can cause problems for fault diagnosis. The problem that arises is: if a component breaks, how should the model be

updated? In some cases this is straightforward: spring breakage in the mass/spring/damper system, for example, is correctly modeled by setting k to 0 in the equation

$$m \cdot x'' + d \cdot x' + k \cdot x = F(t)$$

Other faults present greater difficulty. If the damper seizes up, it may be said that its resistivity has become infinite. Setting d to ∞ , however, does not yield a valid equation. Even the expedient of using a very large number for d may not be useful: simulation and analysis are difficult or impossible with such parameters. As will be evident in the subsequent discussion, the problem is that the equation has been invalidated by the fault.

In order to obtain a valid model of the faulty system, it is necessary to know what entities the attributes (parameters) are attributes of, and what the *role* of these entities is. Lacking this information, if the entity/component breaks, we cannot determine what should happen to its attributes in the equation, or how the equations change.

It should be noted that the concept of role is not easily pinned down. Among other things, the role of a component has to do with the purpose of the system of which it is a part, in particular the way it enables the containing system to perform *its* role. We clearly have a recursive definition here, ending only at the top level, where the role of the entire system is defined in terms of human goals. The role of a wheel of an automobile, for example, is multiple: the wheel enables propulsion, heading control, and ground friction minimization. The role of the car in its entirety, on the other hand, has to do with the human goal of transportation.

In the present context, the term *role* refers to the category of building block a particular entity is in terms of the regnant model ontology. We thus require an ontology that allows us to categorize components of physical systems as one of a finite, well-defined set of building blocks, each of which plays a fixed, well-defined role in the system.

What Ontology is Appropriate?

We have indicated above that a major motivation for the use of quantitative models, in particular quantitative models based on state equations, was the fact that the ontology underlying such models was superior to the alternatives. In particular, the notational system for describing dynamical systems provided by *bond graphs* [R&K] provided many of the required features: a simple but powerful component-oriented ontology, the capability for automatic generation of (quantitative) state equations, and automatic generation of constraint networks that can serve as basis for qualitative reasoning. Most importantly, however, bond graphs allow model updates that correctly reflect faults. Since reasoning from first principles inherently requires a model embodying those principles, this last capability is crucial.

DYNAMICAL SYSTEMS/BOND GRAPH ONTOLOGY

The world according to bond graphs consists of only a few constituents. Fundamental concepts are *effort* and *flow*, which are generalizations of

- force and velocity, torque and rotational speed
- voltage and current
- pressure and fluid flow
- temperature differential and heat flow
- cause and effect, yin and yang, etc.

Among the unexpected consequences of the bond graph notation is the duality between effort and flow it brings to light. Intuitively, concepts such as, say, force and velocity would seem to belong to inherently different categories; formally, however, they are seen to be interchangeable.

Given the concepts of effort and flow, the bond graph ontology provides entities - *components* - to store, transform, and dissipate these basic constructs. Generalized *capacitances* are devices that store effort: springs store mechanical effort (force), capacitors store electrical effort (voltage), surge tanks store fluidic effort (pressure), thermos bottles store thermal effort (heat).

Generalized *inductances* store flow. Physical masses and moving fluids store physical flow, i.e. velocity, in the form of momentum; electrical coils store current. It is interesting to note that there are no thermal inductances.

Generalized *resistances* dissipate power. Examples are provided by shock absorbers, resistors, clogs in pipes, friction, and a multitude of other devices. It is not difficult to find ways to waste energy.

An important component category transforms effort into flow and vice versa. Levers, pulleys, transmissions, transformers, pumps, turbines, gyrators are examples of devices that perform this function in various energy domains.

It should be noted that resistances are actually transformers, a fact that becomes obvious when conservation of energy is considered. Thus resistors, shock absorbers, and the like perform their function by acting as transformers that transform one form of energy, such as electrical or mechanical, into another, which is deemed to exit the system without interesting interactions. The latter form of energy is typically thermal, which is transferred (harmlessly, if all goes well) to the environment. The concept of resistance is thus merely a convenient abbreviation.

Finally, there are *effort sources* and *flow sources*. Effort sources impose an effort on a component; they provide a generalization of motors, batteries, generators, pumps, and similar devices.

Flow sources, as the name indicates, impose a flow. This is a far less intuitive notion than that of effort source, since ostensibly instantaneous imposition of a given flow requires infinite energy. Flow sources are in fact a convenient fiction: we treat something as a flow source if the amount of effort it supplies is essentially infinite with respect to the object to which it is applied. Thus the road imposes a flow on a car wheel; a large engine driving a small auxiliary pump imposes a flow on whatever is being pumped, etc. The concept of flow source did not originate with bond graphs; current sources, for example, are a familiar concept of electrical engineering. As regards bond graphs, however, an important consequence of admitting flow sources is the duality that is thereby produced between effort and flow results.

And that's all! Or almost all. A small number of auxiliary (albeit important) concepts exist, such as signals and modulated transformers. By and large, however, the constructs presented above comprise a *complete* listing of the bond graph ontology. To recapitulate: the effectiveness of bond graphs for our application arises from the following sources:

1. The ontology is quantitative but still noun-oriented. That is, it is possible to algorithmically transform bond graphs into sets of state equations, or if desired, block diagrams or signal flow graphs (or their close relatives, Kuipers diagrams). All of these forms, however, are adjective-oriented: they consist of sets of constraints holding among attributes of entities that have been abstracted away. If we are to reason, however, about the effects that malfunctioning entities will produce, we need to remember at least what these entities are, and what attributes referenced in the constraint sets go with which entities.

2. The ontology provides a very small set of primitives: the number of constructs involved is approximately ten. Alternate device-oriented ontologies do exist; [deK], for example, refers to devices, conduits, and "stuffs" transformed and transported by these. The number of device and stuff types, however, appears to be unlimited. If the building blocks to be used can be invented ad hoc, systematic model construction and processing becomes extremely difficult.

Constitutive Relations

For each component, an effort:flow relationship, called the *constitutive relation*, exists that describes the operation of the component. For electrical resistances, for example, we have the familiar equation

$$V = i \cdot R$$

while mechanical dampers can often be approximated by $F = d \cdot x'$. The general form of resistance constitutive relations is

$$\text{effort} = f_R(\text{flow})$$

i.e. effort is a function of how much flow there is at the present moment.

A similarly general relation governs capacitances. Thus, the formula for an electrical capacitor is

$$V_{\text{capacitor}} = 1/C \int q \, dt$$

A spring is an example of a mechanical capacitance; by Hooke's Law we have

$$F_{\text{spring}} = k \cdot x = k \int \text{velocity} \cdot dt$$

The general form of the capacitance constitutive relation is

$$\text{effort} = f_C(\int \text{flow} \, dt)$$

i.e. effort is a function of how much flow has *accumulated*.

The general constitutive relation for inductances is

$$\text{flow} = f_I(\int \text{effort} \, dt)$$

i.e. flow is a function of how much effort has *accumulated*. Examples are provided by the relations

$$q = \lambda \int V \, dt$$

for electrical current, and

$$\text{velocity} = 1/m \int \text{Force} \, dt$$

for mechanical motion.

Bond graphs are a graphical device for representing this ontology. They express what components are present, and how power flows among these components.

The concept of *power* in this context is used more loosely than is the case in traditional physics. The formal definition, of course is:

$$\text{power} = \text{effort} * \text{flow}$$

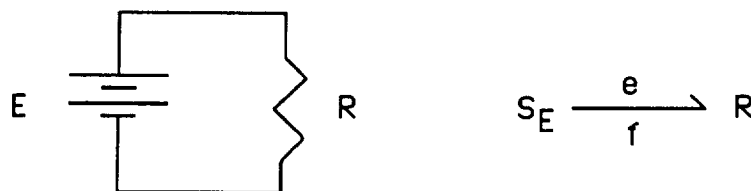
For our purposes, however, the notion of power is often treated less formally. The system under consideration is deemed to contain a variety of types of effort, effort being the activating principle whose resulting activation is termed flow. Thus each component responds to an effort that is imposed on it by manifesting a characteristic kind of flow. Duality considerations require that there also be devices that respond to flow with effort, although strictly speaking this aspect of the modeling process is a convenient fiction.

Dealing with power flow rather than "effort and flow" flow simplifies the modeling process. The effort imposed on a component, and the resulting flow, are bundled into a single graphic element that facilitates the specification of effort and flow sources, destinations, and constraining relationships among them. Furthermore, junctions permit the concise specification of what parts of the system share a common effort, respectively a common flow.

Graphical Constituents of Bond Graphs

Bond graphs consist of nodes corresponding to system components, junctions specifying common effort and common flow constraints, and arcs connecting these elements. These arcs, called *bonds*, are a graphical device denoting *power flow* among components.

Example: A circuit consisting of an effort (voltage) source and a resistance.



The resulting bond graph consists of two nodes corresponding to the voltage source and resistor components, and a bond connecting the two.

The hook on the bond indicates the direction of power flow. Effort and flow variables are associated with the bond; the relation

$$\text{power} = e \cdot f$$

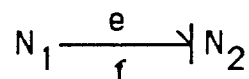
always holds.

Since bonds are the basic constituent of bond graphs, it is useful to examine their meaning more closely. The graphic representation of a bond makes it appear as if some sort of substance were flowing between the constituents it connects, but this is not usually the case. Instead, the meaning of a bond between two constituents is that one of the components is imposing an effort on the other, which is responding with flow. The relationship between the imposed effort and the resulting flow is given by the constitutive relation. Furthermore, the causal stroke (see below) indicates which of the components is imposing the force, and which is responding with flow. It is, however, an error to assume that this flow is the flow of some substance from one of the constituents to the other. Thus, a car's engine imposes an effort on the car, which responds with flow (motion); in no sense, however, is this motion flowing back from the car to the motor.

This raises the question of how the actual flows of materials are represented in a bond graph, if not by bonds. The answer is that such flows have been abstracted away and thus are *not* represented, and that a single bond graph can be the representation of a wide variety of systems. What is represented is what constituents are imposing efforts and flows on each other, and (by means of *junctions*) which elements are subject to identical efforts and flows.

Causality

One of the most important properties of bond graphs is their ability to specify and express causality. Consider the following graph fragment:



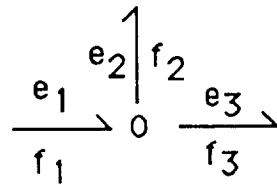
The vertical bar at the end of the bond is called a *causal stroke*, and denotes that component N_1 imposes an effort e on component N_2 , which responds with flow f . In other words, N_2 is a component that inputs effort and outputs flow, while N_1 outputs effort and inputs flow.

We have indicated above the formal duality that exists between effort and flow. Intuitively, however, application of effort to an entity causes it to respond with flow, and thus in the intuitive sense the causality specified by causal strokes is clear: components that output effort act on components that input effort. The resulting flow may act to modulate the magnitude of the effort, but it is difficult to conceive of flow causing effort. The bond graph thus specifies the causality inherent in the modeled system in an explicit and unambiguous manner.

Junctions

Junctions have the function of expressing constraints among multiple components. As stated, the bonds in a bond graph express the flow of power rather than material, and thus do not specify matters such as serial or parallel connectivity in electrical or hydraulic circuits, or physical connections in mechanical systems. Rather, such connectivity is expressed in terms of junctions, which specify the following constraints:

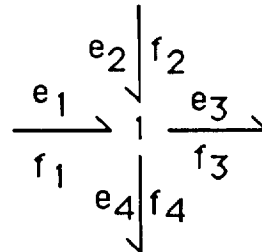
0-junction:
common effort
flows sum to 0



$$e_1 = e_2 = e_3$$

$$f_1 - f_2 - f_3 = 0$$

1-junction:
common flow
efforts sum to 0



$$f_1 = f_2 = f_3 = f_4$$

$$e_2 + e_4 - e_1 - e_3 = 0$$

While the examples show junctions formed of three and four bonds, arbitrarily many more bonds may impinge on junctions. Regardless of the number of bonds, 0-junctions stipulate that all attached bonds share the identical level of effort and that the flows must sum to 0 (incoming flows have positive signs, outgoing ones are negative), while 1-junctions stipulate analogous constraints with the roles of effort and flow reversed.

It is evident that 0-junctions capture the essence of parallel connectivity of components in electrical or hydraulic circuits: such components are necessarily subject to identical effort (pressure or voltage). 1-junctions abstract serial connections: components connected in series clearly will have the same current flowing through each. In the mechanical realm 1-junctions are used to specify that a set of components is constrained to move at the same speed, i.e. that the components are physically coupled together.

Nodes

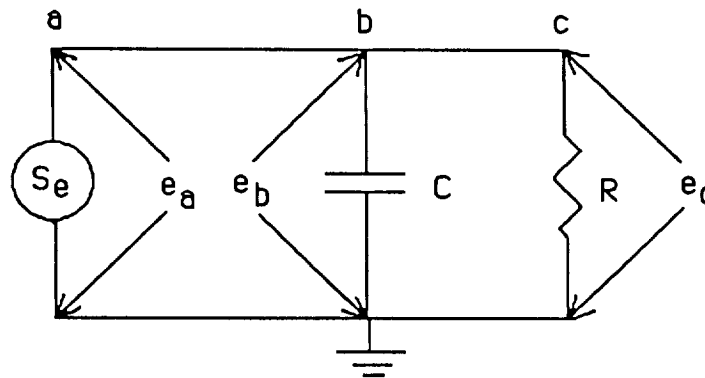
Nodes occurring in bond graphs represent components, or rather the bond graph category of components: capacitances, inductances, etc. The number of bonds impinging on a node is determined by the node category. Capacitances, inductances, resistances, and effort and flow sources have

exactly one bond associated with them; for this reason they are called *one-ports*. Entities such as transformers have two associated bonds, and are, naturally enough, termed *two-ports*.

The idea of categorizing an entity such as an electrical capacitor as a one-port may seem counterintuitive; after all, capacitors have two wires attached to them. This apparent discrepancy disappears when it is recalled once again that bonds are not conduits carrying flows of material such as electrons, but are abstractions denoting power flow. Power either flows into or out of the capacitor, depending on whether it is charging or discharging. By no means, however, can power be deemed to flow in by one lead and out by the other.

As stated above, each component has associated with it a constitutive relation that specifies how the efforts and flows on the bonds associated with the component are related. In the case of an electrical resistance, for example, effort V and flow i are related by the familiar $V = iR$. In fact, as we shall see, the form of a constitutive relation determines the node category just as uniquely as the node category determines the constitutive relation.

We will tie these concepts together by means of an example. Consider the following electrical circuit:

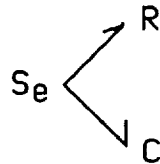


We need to express the following facts:

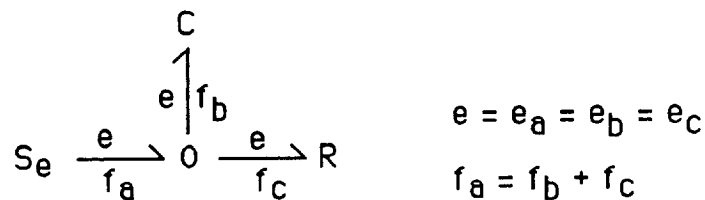
(1) power flows out of voltage source S_e and into C and R

(2) voltages $e_a = e_b = e_c$

Our first attempt might be:



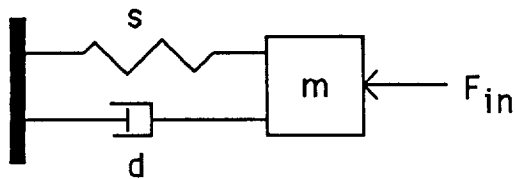
This expresses (1) but not (2). We recall that 0-junctions stipulate equal-effort constraints, which arise from parallel connectivity of components. The following bond graph expresses the necessary:



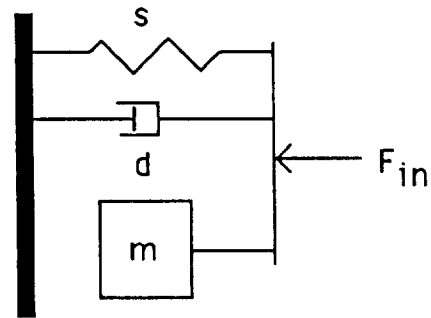
As indicated, the structure of bond graphs expresses constraints among their constituent elements. Conversely, the constraints to be expressed provide extensive guidance in the construction of bond graphs. The following example, involving the construction of a bond graph for the well-known mass/spring/damper system, clarifies this idea.

mass/spring/damper system

physical hookup



conceptual hookup



It is evident that the following elements are present and must be represented:

C (spring s)

S_e

I (mass m)

R (damper d)

The constraints are: velocity f_s of input to spring = velocity f_d of input to damper = velocity f_m of mass

A 1-junction is appropriate for expressing common flow:

$$\begin{array}{c}
 e_s \uparrow \\
 e_{in} \rightarrow \quad 1 \quad \rightarrow e_m \\
 f \quad \quad \quad f \\
 e_d \downarrow
 \end{array}
 \quad f = f_{in} = f_s = f_d = f_m$$

Joining the nodes with these bonds produces

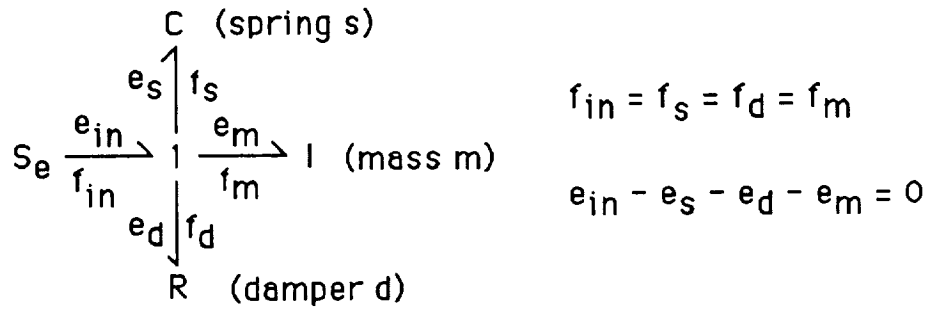


Figure 1

BOND GRAPH-BASED REASONING

Given that we have produced a bond graph model of the system of interest, what can we do with it? As indicated in the preceding section, a bond graph determines a set of constraint equations corresponding to the constraints expressed by the graph's junctions and constitutive relations. Thus, the bond graph of Figure 1 determines the equation set

1. $e_{in} - e_m - e_s - e_d = 0$ (* from the 1-junction *)
2. $f_{in} = f_m = f_s = f_d$ (* from the 1-junction *)
3. $e_{in} = E_e(t)$ (* E_e describes the voltage supplied by S_e *)
4. $e_d = \Phi_d(f_d)$ (* Φ_d is the constitutive relation for the resistance *)
5. $e_s = \Phi_s(\int f_s dt)$ (* Φ_s is the constitutive relation for the capacitance *)
6. $f_m = \Phi_m(\int e_m dt)$ (* Φ_m is the constitutive relation for the inductance *)

Equations (5) and (6) can also be put in derivative form:

- 5'. $e'_s = \Gamma_s(f_s)$
- 6'. $f'_m = \Gamma_m(e_m)$

Equations 1-4, 5', 6' are stated in terms that are immediately translatable to the QSIM paradigm of [Kui], thus establishing the feasibility of qualitative reasoning and qualitative simulation on the basis of bond graphs. This theme will be developed further in a subsequent paper; in the present report we will report on methods of reasoning on the basis of the state equations that can be derived automatically from any bond graph.

Derivation of State Equations

Bond graphs represent dynamical systems, which can be described by state equations having the canonical form

$$x'_1 = F_1(x_1, \dots, x_n, u_1, \dots, u_r)$$

...

$$x'_n = F_n(x_1, \dots, x_n, u_1, \dots, u_r)$$

where the x_i are state variables, and the u_i are input variables. We will sketch the procedure for transforming bond graphs into state equations; details are contained in [R&K].

Sketch of Transformation Procedure

Each inductance has a generalized momentum p associated with it, which is the accumulation (mathematically: the integral) over time of the effort that has acted on it:

$$p = \int e \, dt$$

The constitutive relation for the inductance will have the form

$$f = f_I(p) = f_I(\int e \, dt)$$

E.g. for a mass we have effort $F = ma = mv' = p' = dp/dt$, so flow v is

$$v = \int v' dt = (1/m) \cdot \int p' dt = p/m = f_I(p)$$

Similarly, each capacitance has a generalized displacement q associated with it, which is the integral of the flow to which it has been subjected over time:

$$q = \int f dt$$

The constitutive relation for the capacitance will have the form

$$e = f_C(q) = f_C(\int f dt)$$

For a spring operating under Hooke's Law, for example, we have, for effort F ,

$$F = k \int x' dt = kx = kq = f_C(q)$$

x' is, of course flow, whereas x is displacement q .

The state equations will contain these p 's and q 's as state variables, while the $E(t)$ and $F(t)$ associated with effort and flow sources become input variables. The systematic procedure for deriving canonical-form state equations in p , p' , q , q' from bond graph junction constraints and device constitutive laws is described in [R&K]; the following example gives the flavor of the process.

Example:

For the mass/spring/damper system we have these equations:

From the bond graph junctions:

$$e_{in} = e_m + e_s + e_d \qquad f_{in} = f_m = f_s = f_d$$

The constitutive laws of the devices involved are:

$$R: e_d = d f_d \quad C: e_s = k \int f_s \quad I: f_m = (1/m) \int e_m dt$$

Also, by definition of displacement and momentum,

$$q = \text{displacement} = \int f_s dt, \text{ so } q' = f_s, \text{ and } e_s = k q$$

$$\text{and } p = \text{momentum} = \int e_m dt, \text{ so } p' = e_m$$

Using $e_{in} = e_m + e_s + e_d$ and the constitutive laws, we have

$$E(t) = p' + k q + d p/m$$

and since $q' = f_s = f_m = p/m$, we obtain the state equations

$$p' = -k q - d p/m + E(t)$$

$$q' = p/m$$

REASONING ABOUT MALFUNCTIONS

We have emphasized above that bond graphs provide a noun-oriented (or, in programming parlance, object-oriented) ontology. This orientation yields a major advantage: in a gratifying number of situations, bond graph models can be updated in a systematic manner to reflect malfunctions. In the subsequent discussion we explore what the consequences for the bond graph model are when objects of the modeled system break.

Updating Bond Graph Models to Reflect Malfunctions

Since bond graphs *are* object-oriented, certain components of the physical system may correspond directly to entities in bond graph model. Thus, a generator would play the role of an effort source, while a rivet would enforce common flow, i.e. act as a 1-junction. A turbine is a transformer, a dashpot is a resistance, as is an electrical resistor, as is friction, etc. Those components that do not correspond directly to a bond graph entity will be a component part of such an entity. In the extreme, the component must be a part of the system as a whole, which is modeled by the entire bond graph.

Suppose, then, that a malfunction in component X occurs. The following procedure is used to update the bond graph model.

Step 1:

X either corresponds to a bond graph constituent C,
or
X part_of Y ... part_of Z,

where Z corresponds to a bond graph constituent C.

Update the bond graph to reflect the failure of C.

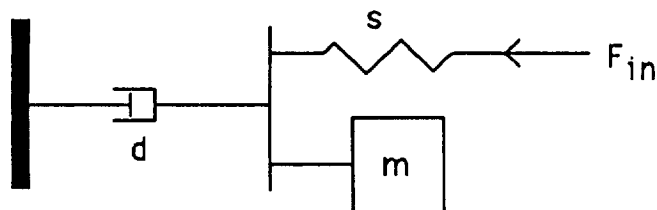
Step 2. Generate new state equations from updated bond graph

Step 3. Prognosticate on basis of updated model

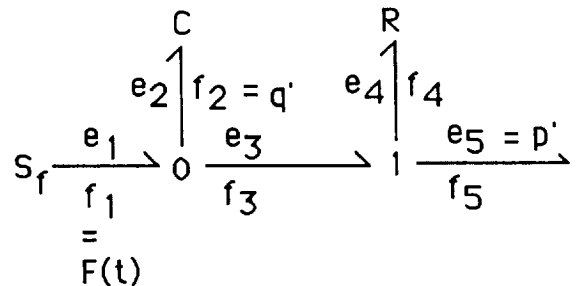
Updating the bond graph to properly reflect a malfunction is a knowledge-based operation: resistors do not fail in the same way as shock absorbers, etc. We have found, however, that in a great majority of cases the malfunction can be modeled by clamping an effort or flow to 0. This observation becomes plausible when we consider what most malfunctions entail: either something that is supposed to move (flow) stops moving, or something that is supposed to be exerting force (effort) on something else stops doing so. The following example illustrates this point.

Example of Malfunction Processing

Consider this mechanical system:



where s is a spring, m a mass, d a resistance, and $F_{in}(t)$ is the input force. This system has the bond graph



Deriving the system's state equations yields:

$$\begin{aligned} q' &= F(t) - p/l \\ p' &= q/C - R \cdot p/l \end{aligned}$$

Now suppose that the dashpot breaks. Draphys examines the symptoms and produces a diagnosis: malfunction in component D-4162AK3. Accessing the component database produces a variety of useful information, including the fact that the component type is dashpot, and that such a component corresponds directly to a resistive element (element "R") in the bond graph.

The database also contains possible failure modes for dashpots of D-4162AK3's type. These are:

seize-up : movement stops (flow clamped at 0)

breakage: force transmission stops (effort clamped at 0)

fluid leakage : complex failed behavior

Generally each failure mode of a component will produce a distinctive set of symptoms; if this is not the case, alternate possibilities must be explored. For our example we will examine the case where the dashpot has seized up.

A seized-up damper has effectively infinite resistance; if the forces on it do not lead to breakage, all movement of the damper shaft ceases.

In terms of the equations of the traditional dynamical systems model, deKleer's confluences [deK], or the processes of Qualitative Process Theory [For], it is not obvious what happens to the intact system as resistance $R \rightarrow \infty$. In terms of the bond graph model, however, we can reason as follows:

flow f_4 through R becomes 0. Thus:

$$f_4 = f_5 = f_3 = 0, \text{ and so}$$

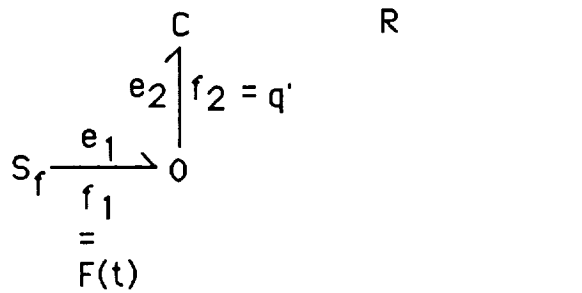
$$e_4 \cdot f_4 = 0 \quad (* \text{ power to resistor } R *)$$

$$e_5 \cdot f_5 = 0 \quad (* \text{ power to inductor } I *)$$

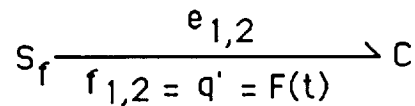
$$e_3 \cdot f_3 = 0 \quad (* \text{ power to the } I/R \text{ serial system } *)$$

Bonds denote power flow, so no power flow between nodes means that no bond exists between these nodes. It is thus evident that the appropriate model update to reflect the damper seize-up is to erase the bonds whose power flow is 0, rather than modifying the constitutive law of the resistance. This insight is a major contribution of the bond graph ontology; the traditional means for reflecting malfunctions has been to relax the constraints imposed by the constitutive relations defining the faulty component.

With these bonds gone, the system becomes



Two-port junctions can be eliminated (see [R&K] for details), producing:



Since $F(t) = f_1 = f_2 = q'$, the updated model has state equation

$$q' = F(t)$$

This is intuitively plausible, since with the dashpot immobilized, all that remains of the system is the flow source driving the capacitance (spring).

Reasoning with Bond Graph Models

A more precise definition of the concept of model than was given above is in order. For our purposes a *model* consists of:

- a set of variables
- a domain over which the variables range, and
- a set of constraints on the variables.

In the fully general case, the constraints are simply theories in the predicate calculus; in practice, however, specialized vocabularies and notations are the rule. Bond graphs, for example, are specified in terms of a domain of nodes, plus constraints that specify bond connectivity among

nodes, the ontological types and constitutive relations associated with the nodes, causality and power flow, and similar information. Bond graph models can be transformed systematically into state equation models, in which the domain is Euclidean n -space, and the constraints take the form of a set of equations.

The preceding definition of models was included to permit a clear distinction between the model itself, and the techniques available for extracting information from the model. For the case of linear state equations, for example, we distinguish between the *model*, which consists of the equation set, and the *information extraction techniques*, usually termed Dynamical Systems Theory, and including techniques such as Laplace Transforms, vector and matrix analysis, and a multitude of similar tools.

Research performed subsequent to the work being reported here has led to a number of techniques for extracting information from bond graphs. These have largely involved transforming bond graphs into qualitative models, particularly Kuipers models, and reasoning on the basis of the transformed system by means reported in the qualitative physics literature. The present report has concerned itself chiefly with model update techniques, and with methods for performing *quantitative* reasoning on the basis of bond graphs.

We have seen that bond graphs can be systematically transformed into systems of state equations in canonical form, and that bond graphs facilitate modification to reflect system malfunctions. The updated model will again be a bond graph, and thus we can obtain state equations representing the faulty system. Work to date on quantitative reasoning with bond graphs has focused on quantitative reasoning on the basis of state equations.

What kinds of quantitative reasoning can be performed using a state equation representation? This is tantamount to asking what the results of dynamical systems theory are, and admits no concise answer. A more appropriate question is: what kinds of quantitative techniques can be

applied to state equation sets to produce information useful to flight crews?

Two categories of information present themselves immediately:

1: given that the fault has been diagnosed and the model updated to reflect it, how will it propagate? and

2: how will the behavior of the aircraft (sub)system be affected?

As it happens, quantitative reasoning is better suited for the second category of question, while qualitative reasoning is appropriate for the first. This is true because functional propagation is an ill-specified phenomenon for which accurate models generally are unavailable, since propagation is often caused by components being driven beyond their design specifications by upstream fault occurrences. In most cases, however, the data and theory required to model component behavior precisely is available only for the component's design envelope. An example is provided by the flight envelope specifications for an aircraft. This specification may stipulate that the aircraft can sustain a "g" loading of 6 g's, with a safety margin of 1.5. The pilot would be ill-advised, however, to assume on this basis that he can safely impose a load of 8.9, but that the aircraft will disintegrate at 9 g. Behavior at design envelope edges typically becomes chaotic, both in the colloquial and mathematical sense. (This is true essentially by definition: the envelope is usually drawn where chaotic behavior begins.)

The upshot of these considerations is that component failure propagation can not generally be predicted with quantitative precision, and that this is an appropriate arena for qualitative techniques. A very general and abstract approach is to extend the techniques incorporated in Draphys to extend the propagation simulation beyond the present time point. This has been accomplished, and is implemented in a PC-based prototype.

Predicting the behavior of the faulted system, given the updated model, is far more appropriate for quantitative techniques. The most obvious information extraction technique is to perform a simulation by numeric

solution of the state (differential) equations. While a flight crew in an emergency situation will not usually have the time required for leisurely perusal of reams of simulation results, a user-friendly graphics-based interface can yield a comprehensible preview of system behavior. It is interesting to note that the information sought from even quantitative models is nonetheless qualitative in many respects; "will g loads reach dangerous levels?" is a typical formulation. An interface that can produce qualitative answers from quantitative data is thus an indispensable adjunct to the simulation.

A wide variety of analytical techniques are relevant as well to extracting qualitative information from state equation models. A detailed exposition of these may be found in [P&L]; we will present an example based on the use of Jacobians to determine the effect of changes in parameters on the equilibrium behavior of the system. The relevance of this technique is based on the fact that the parameters involved are quantities that were designed to remain constant, but may undergo changes induced by faults.

Analysis of Parameter Change Effects

Consider the system of state equations

$$\begin{aligned}x'_1 &= f_1(x_1, \dots, x_n, c_1, \dots, c_n) \\&\quad \text{----} \\x'_n &= f_n(x_1, \dots, x_n, c_1, \dots, c_n)\end{aligned}$$

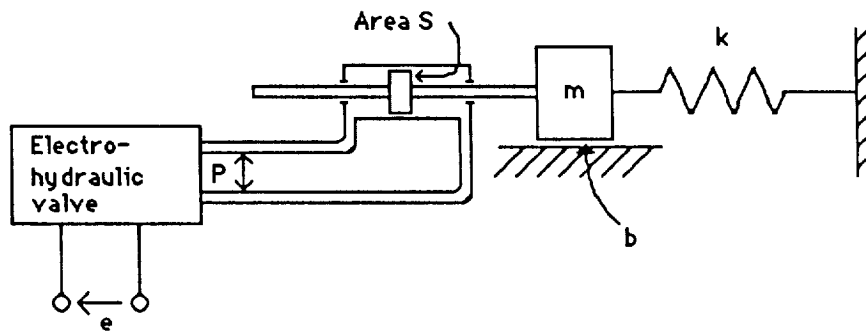
where the c_i denote any parameters occurring in the equations besides the x_i variables such as coefficients and input variables. We pose the query: if c_j increases, what happens to the equilibrium level of x_i ?

Assuming our set of state equations is linear:
The formula

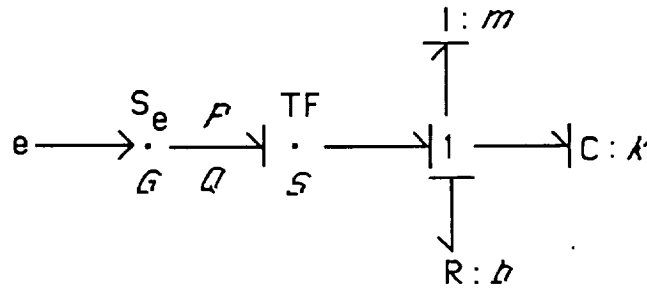
$$\frac{\partial x_i}{\partial c_j} = \frac{\begin{vmatrix} a_{11} & \dots & a_{1i-1} & -\partial f_1 / \partial c_j & a_{1i+1} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{ni-1} & -\partial f_n / \partial c_j & a_{ni+1} & \dots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix}}$$

serves to determine such relationships [P&L].

We will illustrate a typical qn-model-based reasoning technique in light of an example: a physical system consisting of a servomechanism constructed from an electrohydraulic valve, which supplies pressure proportional to a control voltage. This hydraulic pressure P is applied to a ram of area S which moves a mass m represented by a spring constant k and a damper (friction) of constant b [R&K, p. 420].



By applying techniques detailed in [R&K] we obtain the bond graph model of this system:



This bond graph yields the following equations:

$$\dot{q} = p/m$$

$$p' = Ge/S - kq - p b/m$$

What is the effect of a change in the gain G on displacement q at equilibrium? We have:

$$\frac{\partial q}{\partial G} = \frac{\begin{vmatrix} 1/m & 0 \\ -b/m & -e/S \end{vmatrix}}{\begin{vmatrix} 1/m & 0 \\ -b/m & -k \end{vmatrix}} = e/(S \cdot k)$$

Transformer and damper constants S and k are necessarily > 0 ; therefore q varies directly with G .

Superficially these conclusions appear identical to the sort of information that could be extracted from a qualitative model, particularly a Kuipers model. In fact, however, significant differences exist. Thus, the monotonic relationships given above refer to *steady-state* behavior rather than direct, presumably transient, influences of one variable on another. In most cases it is steady-state behavior that is of interest, which is difficult to determine from qualitative models.

In addition, the above relationships are clearly quantitative. Thus, if G

changes by $p\%$, q will change by $p \cdot e / (S \cdot k)\%$.

Many more qualitative answers can be derived from quantitative (state equation) models, including oscillation and long-term behavior. For the sake of uniformity we rewrite the hydraulic system equations as:

$$q' = 0 \cdot q + (1/m) \cdot p + 0 \cdot e(t)$$

$$p' = -k \cdot q - (b/m) \cdot p + (G/S) \cdot e(t)$$

Applying the Laplace transform, we obtain transformed equations

$$\begin{aligned} sQ(s) - q(0) &= 0 \cdot Q(s) + (1/m) \cdot P(s) + 0 \cdot E(s) \quad \text{and} \\ sP(s) - p(0) &= -k \cdot Q(s) - (b/m) \cdot P(s) + (G/S) \cdot E(s) \end{aligned}$$

Solving for $Q(s)$ and $P(s)$, we obtain

$$Q(s) = \frac{\begin{vmatrix} q(0) & -1/m \\ p(0) + (G/S)E(s) & s + b/m \end{vmatrix}}{\begin{vmatrix} s & -1/m \\ -k & s + b/m \end{vmatrix}} = \frac{q(0)(s + b/m) + (1/m)(p(0) + (G/S)E(s))}{s^2 + (b/m)s + k/m}$$

$$P(s) = \frac{\begin{vmatrix} s & q(0) \\ -k & p(0) + (G/S)E(s) \end{vmatrix}}{\begin{vmatrix} s & -1/m \\ -k & s + b/m \end{vmatrix}} =$$

$$\frac{s(p(0) + (G/S)E(s)) + k \cdot q(0)}{s^2 + (b/m)s + k/m}$$

The roots of the characteristic polynomial $s^2 + (b/m)s + k/m$ are

$$s = \frac{-b/m \pm \sqrt{(b/m)^2 - 4(k/m)}}{2}$$

If the discriminant $(b/m)^2 - 4(k/m)$ is positive, the roots are real and the system will not oscillate. If it is negative, the system will undergo damped oscillations [R&K, Ch. 5].

To determine long-term steady-state behavior, we use the Final Value Theorem of the Laplace transform. Thus we have

$$q(t \rightarrow \infty) = \lim_{s \rightarrow 0} sQ(s) = (G/k \cdot S) \cdot \lim_{s \rightarrow 0} sE(s)$$

$$p(t \rightarrow \infty) = \lim_{s \rightarrow 0} sP(s) = m \cdot G / (k \cdot S) \cdot \lim_{s \rightarrow 0} s^2 \cdot E(s)$$

If $e(t) = \text{constant } e$, then $E(s) = e/s$, so

$$q(t \rightarrow \infty) = (G/S) \cdot \lim_{s \rightarrow 0} s \cdot e/s = Ge / (k \cdot S), \text{ and}$$

$$p(t \rightarrow \infty) = m \cdot G / (k \cdot S) \cdot \lim_{s \rightarrow 0} s^2 \cdot e/s = 0$$

Thus for constant e , the block moves to a fixed location and stops.

CONCLUSION

The work described in this report has addressed the problem of determining the behavior of physical systems subsequent to the occurrence of malfunctions. It was established that while it was reasonable to assume that the most important fault behavior modes of primitive components and simple subsystems could be known and predicted, interactions within composite systems reached levels of complexity that precluded the use of traditional rule-based expert system techniques. Reasoning from first principles, i.e. on the basis of causal models of the physical system, was required.

The first question that arises is, of course, how the causal information required for such reasoning should be represented. While the work on qualitative physics exemplified by the papers in [QR] represents an obvious starting point, it soon became apparent that the modeling techniques set forth there were not well suited to the requirements of representing malfunctions of components in continuous physical systems. Models based on [deK] and [Kui], like traditional equation-based quantitative models, consist of sets of constraints among attributes of entities that have been abstracted away. It is clearly difficult to update a model to reflect the failure of entities that are not represented in the model. Forbus' QP models, on the other hand, do explicitly reference physical entities in their *Individuals* and *Preconditions* sections; these representations, however, are too haphazard and unsystematic to make it clear what new entities, propagation paths, and processes come into being when a malfunction occurs.

The bond graphs presented in this report occupy a position intermediate between qualitative and quantitative models, allowing the automatic derivation of Kuipers-like qualitative constraint models as well as state equations. Their most salient feature, however, is that entities corresponding to components and interactions in the physical system are explicitly represented in the bond graph model, thus permitting systematic model updates to reflect malfunctions. We have shown how this is done, as well as presenting a number of techniques for obtaining qualitative

information from the state equations derivable from bond graph models.

It is typical of research projects such as the present one that certain insights unifying a variety of aspects of the research are reached only at the very conclusion of the work. One such insight is the fact that one of the most important advantages of the bond graph ontology is the highly *systematic* approach to model construction it imposes on the modeler, who is forced to classify the relevant physical entities into a small number of categories, and to look for two highly specific types interactions among them. The systematic nature of bond graph model construction facilitates the process to the point where the guidelines are sufficiently specific to be followed by modelers who are not domain experts. As a result, models of a given systems constructed by different modelers will have extensive similarities. Furthermore, the process is sufficiently "top-down" to allow at least coarse-grained models to be created if detail is unavailable. The successor report to this one will illustrate these points by presenting the construction of a jet engine model (correct, to the given level of detail), by a modeler whose knowledge of such engines is quite limited.

We conclude by pointing out that the aforementioned ease of updating bond graph models to reflect malfunctions is a manifestation of the systematic nature of bond graph construction, and the regularity of the relationship between bond graph models and physical reality.

In a subsequent report we will consider the process of bond graph model construction and updating in greater detail, as well as showing how such models can be automatically transformed not only into qualitative models, but also into continuous simulation models. This capability provides an interface with earlier work, which applied predicate transformer techniques to continuous simulations. In addition, we will show how bond graphs can be integrated into traditional AI representations such as frames and semantic nets, thus allowing automatic interpretation of the predictions produced by the embedded models.

REFERENCES

- [Abb] Abbott, K., *Robust Fault Diagnosis of Physical Systems in Operation*, Ph. D. Dissertation, Computer Science Department, Rutgers University, New Brunswick, NJ, May 1990.
- [QR] Bobrow, D. (ed.), *Qualitative Reasoning about Physical Systems*, The MIT Press, Cambridge, MA, 1985.
- [deK] deKleer, J., & J. Brown, *A Qualitative Physics Based on Confluences*, in *Qualitative Reasoning about Physical Systems*, D. Bobrow, (ed.) The MIT Press, Cambridge, MA, 1985.
- [For] Forbus, K., *Qualitative Process Theory*, *ibid.*
- [Kui] Kuipers, B., *Commonsense Reasoning about Causality*, *ibid.*
- [P&L] Puccia, C., & R. Levins, *Qualitative Modeling of Complex Systems*, Harvard University Press, Cambridge, MA, 1985.
- [R&K] Rosenberg, R., & D. Karnopp, *Introduction to Physical System Dynamics*, McGraw-Hill, 1983.
- [Sch] Schutte, P., *Real-time Fault Monitoring for Aircraft Applications using Qualitative Simulation and Expert Systems*, in Proc. of the AIAA Computers in Aerospace VII Conference, Monterey, CA, October 1989.